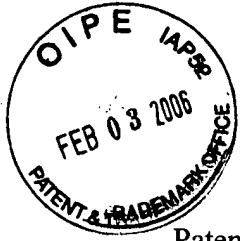


10/616 737

C of D



**PATENT**  
Attorney Docket No. 56507

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Patent No. 6,988,106

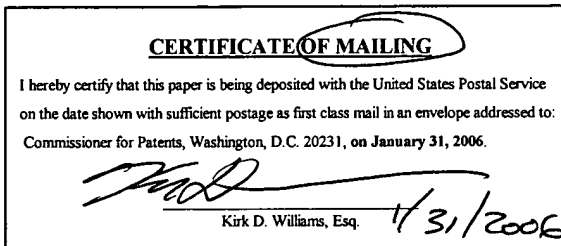
Confirmation No. 2761

Issued: January 17, 2006

Name of Patentee: Enderwick et al.

**Certificate**  
**FEB 08 2006**  
**of Correction**

Patent Title: STORING AND SEARCHING A  
HIERARCHY OF ITEMS OF PARTICULAR  
USE WITH IP SECURITY POLICIES AND  
SECURITY ASSOCIATIONS



**REQUEST FOR CERTIFICATE OF CORRECTION OF  
PATENT FOR PATENT OFFICE MISTAKE (37 C.F.R. § 1.322)**

Attn: Certificate of Correction Branch  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

It is requested that a Certificate of Correction be issued to correct Office mistakes found the above-identified patent. Attached hereto is a Certificate of Correction which indicates the requested correction. For your convenience, also attached are copies of selected pages (a) from the issued patent with errors highlighted, and (b) from the original application as filed July 9, 2003.

FEB 9 2006


In re US Patent No. 6,988,106

It is believed that there is no charge for this request because applicant or applicants were not responsible for such error, as will be apparent upon a comparison of the issued patent with the application as filed or amended. However, the Assistant Commissioner is hereby authorized to charge any fee that may be required to Deposit Account No. 501430.

Respectfully submitted,  
**The Law Office of Kirk D. Williams**

Date: January 31, 2006

By



Kirk D. Williams, Reg. No. 42,229  
One of the Attorneys for Applicants  
CUSTOMER NUMBER 26327  
The Law Office of Kirk D. Williams  
1234 S. OGDEN ST., Denver, CO 80210  
303-282-0151 (telephone), 303-778-0748 (facsimile)

FEB 9 2006

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,988,106  
DATED : January 17, 2006  
INVENTOR(S) : Enderwick et al.

It is certified that error(s) appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Face of the Patent, Patent Title (54), replace "STRONG" with -- STORING --

Col. 1, line 31, replace " Internet. Protocol" with -- Internet Protocol --

Col. 1, line 67, replace "bidirectional" with -- bi-directional --

Col. 4, line 42, replace "internet" with -- Internet --

Col. 13, line 54, replace "first, TCAM" with -- first, the TCAM --

MAILING ADDRESS OF SENDER:

Kirk D. Williams, Reg. No. 42,229  
Customer No. 26327  
The Law Office of Kirk D. Williams  
1234 S. Ogden Street, Denver, CO 80210

PATENT NO. 6,988,106  
No. of additional copies

⇒ NONE (0)

FEB 9 2006



US006988106B2

(12) **United States Patent**  
**Enderwick et al.**

(10) Patent No.: **US 6,988,106 B2**  
 (45) Date of Patent: **Jan. 17, 2006**

(54) **STRONG AND SEARCHING A HIERARCHY OF ITEMS OF PARTICULAR USE WITH IP SECURITY POLICIES AND SECURITY ASSOCIATIONS**

(75) Inventors: **Thomas Jeffrey Enderwick**, San Jose, CA (US); **Henry Kin-Chuen Kwok**, Fremont, CA (US); **Ashwath Nagaraj**, Los Altos, CA (US)

(73) Assignee: **Cisco Technology, Inc.**, San Jose, CA (US)

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 35 days.

(21) Appl. No.: **10/616,737**

(22) Filed: **Jul. 9, 2003**

(65) **Prior Publication Data**

US 2005/0010612 A1 Jan. 13, 2005

(51) Int. Cl. **G06F 17/30** (2006.01)

(52) U.S. Cl. **707/100**

(58) Field of Classification Search **707/3, 707/5, 9, 200, 201, 100**  
 See application file for complete search history.

(56) **References Cited**

#### U.S. PATENT DOCUMENTS

3,648,254 A	3/1972	Beausoleil	365/49
4,296,475 A	10/1981	Nederlof et al.	711/108
4,791,606 A	12/1988	Threewitt et al.	365/49
4,996,666 A	2/1991	Duluk, Jr.	365/49
5,339,076 A	8/1994	Jiang	341/51
5,383,146 A	1/1995	Threewitt	365/49
5,404,482 A	4/1995	Stamm et al.	711/145
5,428,565 A	6/1995	Shaw	365/49
5,440,715 A	8/1995	Wyland	711/108
5,450,351 A	9/1995	Heddes	365/49
5,684,954 A	11/1997	Kaiserswerth et al.	709/236

5,802,567 A	9/1998	Liu et al.	711/133
5,841,874 A	11/1998	Kempke et al.	713/160
5,852,569 A	12/1998	Srinivasan et al.	365/49
5,956,336 A	9/1999	Loschke et al.	370/392
5,978,885 A	11/1999	Clark, II	711/108
6,038,560 A *	3/2000	Wical	707/5
6,041,389 A	3/2000	Rao	711/108
6,047,369 A	4/2000	Colwell et al.	712/217
6,069,573 A	5/2000	Clark, II et al.	341/50
6,081,440 A	6/2000	Washburn et al.	365/49
6,134,135 A	10/2000	Andersson	365/49

(Continued)

#### OTHER PUBLICATIONS

Zao et al., Domain Based Internet Security Policy Management, DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings vol. 1, Jan. 25-27, 2000 Page(s): 41-53.\*

(Continued)

Primary Examiner—Greta Robinson

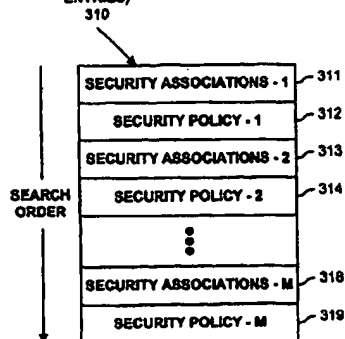
(74) Attorney, Agent, or Firm—The Law Office of Kirk D. Williams

(57) **ABSTRACT**

Mechanisms for storing and searching a hierarchy of items are disclosed which may be particularly useful for implementing security policies and security associations, such as, but not limited to Internet Protocol security (IPsec). A hierarchy of items is stored in a search priority order. Multiple element definitions and groups of elements are identified. Representations of the element definitions and elements are stored in a prioritized searchable data structure in decreasing search priority such that representations of each particular element definition is stored after representations of a set of particular elements associated with the particular element definition and before representations of lower priority element definitions and their associated elements. The element definitions may include Internet Protocol security policies and the elements may include Internet Protocol security associations. The searchable data structure may include an associative memory or a plurality of associative memory entries.

16 Claims, 17 Drawing Sheets

PRIORITIZED SEARCHABLE  
 DATA STRUCTURE  
 (E.G., ASSOCIATIVE MEMORY  
 ENTRIES)



FEB 9 2006

1

# STRONG AND SEARCHING A HIERARCHY OF ITEMS OF PARTICULAR USE WITH IP SECURITY POLICIES AND SECURITY ASSOCIATIONS

## TECHNICAL FIELD

One embodiment of the invention especially relates to communications and computer systems; and more particularly, one embodiment relates to storing and searching a hierarchy of items which may be particularly useful for implementing security policies and security associations, such as, but not limited to Internet Protocol security (IPsec) in routers, packet switching systems, computers, and/or other devices.

## BACKGROUND

The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology. Increasingly, public and private communications networks are being built and expanded using various packet technologies, such as Internet Protocol (IP).

A security architecture for the Internet Protocol (IPsec) is defined in S. KENT and R. ATKINSON, "Security Architecture for IP," RFC 2401, November 1998, which is hereby incorporated by reference.

An IPsec implementation operates in a host or a security gateway environment, affording protection to IP traffic. The protection offered is based on requirements defined by a Security Policy Database (SPD) established and maintained by a user or system administrator, or by an application operating within constraints established by either of the above. In general, packets are selected for one of three processing modes based on IP and transport layer header information matched against entries in the database. Each packet is either afforded IPsec security services, discarded, or allowed to bypass IPsec, based on the applicable database policies.

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. IPsec can be used to protect one or more "paths" between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The set of security services that IPsec can provide includes access control, connectionless integrity, data origin authentication, rejection of replayed packets (a form of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. Because these services are provided at the IP layer, they can be used by any higher layer protocol, e.g., TCP, UDP, ICMP, BGP, etc.

IPsec packet classification is specified as a two-layer hierarchy: the relevant security policy (SP) must be found first out of an ordered list of SPs, and then within the context of the located SP, the correct security association (SA) must be found. A security association is a simplex "connection" that affords security services to the traffic carried by it. To secure typical bidirectional communication between two

2

hosts or between two security gateways, two security associations (one in each direction) are required. A security association is uniquely identified by a triple consisting of a Security Parameter Index (SPI), an IP Destination Address, and a security protocol identifier. In principle, the destination address may be a unicast address, an IP broadcast address, or a multicast group address. The set of security services offered by an SA depends on the security protocol selected, the SA mode, the endpoints of the SA, and on the election of optional services within the protocol. For example, one security protocol provides data origin authentication and connectionless integrity for IP datagrams.

The IP datagrams transmitted over an individual SA are afforded protection by exactly one security protocol. Sometimes a security policy may call for a combination of services for a particular traffic flow that is not achievable with a single SA. In such instances it will be necessary to employ multiple SAs to implement the required security policy. The term "security association bundle" or "SA bundle" is applied to a sequence of SAs through which traffic must be processed to satisfy a security policy. The order of the sequence is defined by the policy. (Note that the SAs that comprise a bundle may terminate at different endpoints. For example, one SA may extend between a mobile host and a security gateway and a second, nested SA may extend to a host behind the gateway.)

RFC 2401 defines that there are two nominal databases in the IPsec general model, with these two databases being the security policy database (SPD) and the security association database (SAD). The former specifies the policies that determine the disposition of all IP traffic inbound or outbound from a host, security gateway, or BITS or BITW IPsec implementation. The latter database contains parameters that are associated with each (active) security association. This section also defines the concept of a selector, a set of IP and upper layer protocol field values that is used by the security policy database to map traffic to a policy, i.e., an SA (or SA bundle).

Each interface for which IPsec is enabled requires nominally separate inbound vs. outbound databases (SAD and SPD), because of the directionality of many of the fields that are used as selectors. Typically there is just one such interface, for a host or security gateway (SG). Note that an SG would always have at least two interfaces, but the "internal" one to the corporate net, usually would not have IPsec enabled and so only one pair of SADs and one pair of SPDs would be needed. On the other hand, if a host had multiple interfaces or an SG had multiple external interfaces, it might be necessary to have separate SAD and SPD pairs for each interface.

Ultimately, a security association is a management construct used to enforce a security policy in the IPsec environment. Thus, an essential element of SA processing is an underlying Security Policy Database (SPD) that specifies what services are to be offered to IP datagrams and in what fashion. The form of the database and its interface are outside the scope of RFC 2401. However, RFC 2401 does specify certain minimum management functionality that must be provided, to allow a user or system administrator to control how IPsec is applied to traffic transmitted or received by a host or transiting a security gateway.

The SPD must be consulted during the processing of all traffic (inbound and outbound), including non-IPsec traffic. In order to support this, the SPD requires distinct entries for inbound and outbound traffic. The SPD contains an ordered list of policy entries. Each policy entry is keyed by one or

②  
Internet  
Protocol  
(no period  
between)

③  
bi-directional

3

more selectors that define the set of IP traffic encompassed by this policy entry. One can think of this as separate SPDs (inbound vs. outbound). In addition, a nominally separate SPD must be provided for each IPsec-enabled interface. A SPD must discriminate among traffic that is afforded IPsec protection and traffic that is allowed to bypass IPsec. This applies to the IPsec protection to be applied by a sender and to the IPsec protection that must be present at the receiver. For any outbound or inbound datagram, three processing choices are possible: discard, bypass IPsec, or apply IPsec. The first choice refers to traffic that is not allowed to exit the host, traverse the security gateway, or be delivered to an application at all. The second choice refers to traffic that is allowed to pass without additional IPsec protection. The third choice refers to traffic that is afforded IPsec protection, and for such traffic the SPD must specify the security services to be provided, protocols to be employed, algorithms to be used, etc.

In each IPsec implementation there is a nominal security association database, in which each entry defines the parameters associated with one SA. Each SA has an entry in the SAD. For outbound processing, entries are pointed to by entries in the SPD. Note that if an SPD entry does not currently point to an SA that is appropriate for the packet, the implementation creates an appropriate SA (or SA Bundle) and links the SPD entry to the SAD entry. For inbound processing, each entry in the SAD is indexed by a destination IP address, IPsec protocol type, and SPI. The following parameters are associated with each entry in the SAD. This description does not purport to be a MIB, but only a specification of the minimal data items required to support an SA in an IPsec implementation.

FIG. 1 illustrates a prior art implementation based on RFC 2401 for processing an outbound packet. Processing begins with process block 100, and proceeds to process block 102, wherein a database lookup operation is performed in the security policy database based on the packet to identify the corresponding security policy. If no policy is found as determined in process block 104, then the packet is dropped in process block 106, and processing is complete as indicated by process block 108. Otherwise, in process block 110, a second lookup operation is performed based on the packet, this time in the security association database corresponding to the security policy identified in the previous lookup operation. As determined in process block 112, if a corresponding security association is not located, then in process block 114, the security association is added to the corresponding security association database. In process block 116, the packet is processed according to the corresponding security association. Processing is complete as indicated by process block 118.

RFC 2401 defines a two-step process for performing lookup operations to in order to identify a SA associated with a packet, i.e., by first performing a lookup in a security policy database and then, performing a subsequent second lookup operation based on the identified security policy to identify the corresponding security association). Especially as packet rates and then number of packets to be processed by a packet processor increases, this two-stage lookup process can be limiting. Desired is a new way of performing IPsec identification operations.

### SUMMARY

Disclosed are, inter alia, methods, apparatus, data structures, computer-readable medium, mechanisms, and means for storing and searching a hierarchy of items which

4

may be particularly useful for implementing security policies and security associations, such as, but not limited to Internet Protocol security (IPsec) in routers, packet switching systems, computers, and/or other devices.

One embodiment stores a hierarchy of items in a search priority order. Multiple element definitions and groups of elements are identified. Representations of the element definitions and elements are stored in a prioritized searchable data structure in decreasing search priority such that representations of each particular element definition is stored after representations of a set of particular elements associated with the particular element definition and before representations of lower priority element definitions and their associated elements. In one embodiment, the element definitions include Internet Protocol security policies and the elements include Internet Protocol security associations. In one embodiment, the searchable data structure includes an associative memory or a plurality of associative memory entries. In one embodiment, an element definition or element corresponding to a range of values is split into multiple entries. In one embodiment, the hierarchy includes more than two levels, and the element definitions and groups of elements are just two of the more than two levels.

One embodiment maintains a data structure for an identified ordered list of Internet Protocol security policies. Ordered associative memory entries associated with the ordered list of Internet Protocol security policies are programmed into one or more associative memories. Corresponding context memory entries associated with the ordered list of Internet Protocol security policies are programmed into one or more context memories. An associative memory lookup operation is performed on the ordered associative memory entries based on a received packet to identify a particular associative memory entry location. A lookup operation is performed on the context memory based on the particular associative memory entry location to identify a particular Internet Protocol security policy of the ordered list of Internet Protocol security policies. A particular security association entry based on the received packet is added to the ordered associative memory entries, the particular security association entry corresponding to the particular Internet Protocol security policy, and the particular security association entry being added to the ordered associative memory entries prior to the particular associative memory entry location and after other security policy entries of the ordered list of Internet Protocol security policies located prior to the particular associative memory entry location.

### BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIG. 1 illustrates a prior art implementation of IPsec;

FIG. 2A is a block diagram illustrating one embodiment for storing and searching a hierarchy of items;

FIG. 2B is a block diagram illustrating one embodiment for storing and searching a hierarchy of items;

FIG. 3A is a block diagram illustrating a prioritized searchable data structure used in one embodiment;

FIG. 3B is a block diagram illustrating a prioritized searchable data structure used in one embodiment;

FIG. 3C is a block diagram illustrating a prioritized searchable data structure used in one embodiment;

④  
Internet

FEB 9 2006

13

one embodiment, two security association databases are used to enhance performance. Outbound security processor 442 processes each outbound packet by first extracting the five selectors specified in RFC 2401, and then performing a search for a match in TCAM 424. If a match is found, outbound security processor 442 indexes the context array using the index of the matched TCAM entry included in lookup results 433. The context array entry indicates whether the TCAM match corresponds to a matching SA or SP. If it is a SP, the context array also consists of the appropriate action for packet matching that SA. If it is a SA, the context array contains the index into the SAD for the corresponding SA. There is only one data structure of outbound SA.

FIG. 5A illustrates associative memory entries used in one embodiment. As shown, TCAM entry 500 includes a source address field 501, a destination address field 502, a source port field 503, a destination port field 504, a protocol type field 505, a service indication field 506, an entry type field 507 to indicate whether the entry is a SA or SP entry, and an implementation specific field 508. Note, one embodiment sets the mask field to don't care in field 507 if the entry corresponds to a service policy because every search is performed on the SPD (e.g., on all SP entries). By not masking out the value when the entry corresponds to an SA, then either all entries can be searched or only SPs can be searched. Thus, global mask register-0 510 has bits set to match in fields 511–516 and to ignore (i.e., don't care) in fields 517–518. Thus, using global mask register-0 510 in a search will cause both SP and SA entries to be searched. Global mask register-1 520 has bits set to match in fields 521–527 and to ignore (i.e., don't care) in field 528. Thus, using global mask register-1 520 in a search with the lookup word specifying SP entry types, a search will cause only SP entries to be searched. Note, the use of block masks are described in Ross et al., "Block Mask Ternary CAM," U.S. Pat. No. 6,389,506, issued May 14, 2002, which is hereby incorporated by reference.

FIG. 5B illustrates a process used in one embodiment for generating multiple associative memory entries for a corresponding range of values. Some applications desire to match on a range of values (e.g., source port number 72–83).

Because TCAMs do not support arbitrary sets or ranges as selection criteria, the splitter is required to perform any required entry expansion. For example, implementing the destination port ranges <25 and >25 requires splitting a single entry into sixteen entries. FIG. 5B illustrates pseudo code of a mechanism used in one embodiment to split entries into multiple entries. The splitter converts a SP specified in a range-set format into a SP specified in an expanded form using a collection of matching values and don't-care mask. For example, support a range of 1 to 15 becomes 4 sets of (matching values, don't care mask): (0x1, 0xe), (0x2, 0xd), (0x4, 0xb), and (0x8, 0x7). As shown, first, TCAM entry d...d is checked to see if it matches a subset of the values covered by the range. If not, then the process is repeated with 0d...d and 1d...d. This happens recursively (using the stacks—not function recursion). Branches are trimmed when the entry being tested matches a disjoint set of values. Entries are saved when they match a subset of the values matched by the range. Entries that match overlapping sets are split and pushed onto the work stack.

FIG. 6A illustrates a process used in one embodiment for processing an inbound packet. Processing begins with process block 600, and proceeds to process block 602, wherein a packet is received. As determined in process block 604, if the packet is marked as conforming to IPsec, then in process

14

block 606 the packet is processed, and processing is completed as indicated by process block 619. Otherwise, in process block 610, a lookup word is generated based on the received packet (e.g., with fields in accordance to those stored in the associative memory or other implementations of the data structure). In process block 612, a lookup operation is initiated and performed in the associative memory using the lookup word and a global mask register such that only SP entries are searched. The lookup result is received and a lookup operation based on the result is performed in the context memory in process block 614. Then, in process block 616, the packet is processed according to the action identified in the context memory. Processing is complete as indicated by process block 619.

FIG. 6B illustrates a process used in one embodiment for processing an outbound packet. Processing begins with process block 640, and proceeds to process block 642, wherein a packet is received. Next, in process block 644, a lookup word is generated based on the received packet. In process block 646, a lookup operation is initiated and performed in the associative memory using the lookup word and a global mask register such that both SP and SA entries are searched. The lookup result is received and a lookup operation based on the result is performed in the context memory in process block 648. As determined in process block 650, if the entry matched corresponds to an SA entry, then in process block 652, the action to perform is identified in the SAD based on the lookup result retrieved from the context memory, and the packet is processed according to the identified action. Otherwise, in process block 660, the packet is processed according to the action identified by the context memory; and in process block 662, a security access entry is added to the SAD and the associative and context memories are updated accordingly. Processing is complete as indicated by process block 669.

FIG. 7 illustrates a process used in one embodiment for adding an entry to an ordered list of associative memory entries. Processing begins with process block 700, and proceeds to process block 702, wherein an associative memory or other prioritized searchable data structure update request is identified. Next, in process block 704, the partition and possibly the exact location(s) to add one or more entries are identified. As determined in process block 706, if there is space to add the one or more entries in the identified partition, then the entries are added in process block 712. Otherwise, space for the new entries is made (or attempted to be made) in process block 708. As determined in process block 710, if this expansion of the partition was successful, then the entries are added in process block 712. Otherwise, there is no room for the entries and an error condition is generated. Processing is complete as indicated by process block 714.

FIGS. 8A–D and 9A–D illustrate processes used in one embodiment for expanding partitions and redistributing space allocated to partitions. Note, these processes may call each in a recursive or other fashion to expand/shrink partitions to redistribute the free space among partitions. One embodiment attempts to maintain an even distribution of free space (or something approximating such) across all partitions to minimize the amount of adjusting to be performed in adding one or more entries to a partition. By maintaining an approximate even distribution of free space among partitions, a single insert of an element or element definition (which may include one or more associative memory entries) can be quickly performed and limits the worst-case insertion time, which is important for applications with high update rates. Note, one embodiment does not

(5)  
first, the  
TCAM

From Original Application filed July 9, 2003

**STORING AND SEARCHING A HIERARCHY OF ITEMS OF PARTICULAR  
USE WITH IP SECURITY POLICIES AND SECURITY ASSOCIATIONS**

1

**TECHNICAL FIELD**

5 One embodiment of the invention especially relates to communications and computer systems; and more particularly, one embodiment relates to storing and searching a hierarchy of items which may be particularly useful for implementing security policies and security associations, such as, but not limited to Internet Protocol security (IPsec) in routers, packet switching systems, computers, and/or other devices.

10

**BACKGROUND**

The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology. Increasingly, public and private communications networks are being built and expanded using various packet technologies, such as Internet Protocol (IP).

20 A security architecture for the Internet Protocol (IPsec) is defined in. S. KENT and R. ATKINSON, "Security Architecture for IP," RFC 2401, November 1998, which is hereby incorporated by reference.

2

An IPsec implementation operates in a host or a security gateway environment, affording protection to IP traffic. The protection offered is based on requirements defined by a Security Policy Database (SPD) established and maintained by a user or system administrator, or by an application operating within constraints established by either of the above. In general, packets are selected for one of three processing modes based on IP and transport layer header information matched against entries in the database. Each



From Original Application filed July 9, 2003

packet is either afforded IPsec security services, discarded, or allowed to bypass IPsec, based on the applicable database policies.

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in  
5 place any cryptographic keys required to provide the requested services. IPsec can be used to protect one or more "paths" between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The set of security services that IPsec can provide includes access control, connectionless integrity, data origin authentication, rejection of replayed packets (a form of partial sequence integrity), confidentiality  
10 (encryption), and limited traffic flow confidentiality. Because these services are provided at the IP layer, they can be used by any higher layer protocol, e.g., TCP, UDP, ICMP, BGP, etc.

IPsec packet classification is specified as a two-layer hierarchy: the relevant security policy (SP) must be found first out of an ordered list of SPs, and then within the  
15 context of the located SP, the correct security association (SA) must be found. A security association is a simplex "connection" that affords security services to the traffic carried by it. To secure typical bi-directional communication between two hosts or between two security gateways, two security associations (one in each direction) are required. A security association is uniquely identified by a triple consisting of a Security Parameter  
20 Index (SPI), an IP Destination Address, and a security protocol identifier. In principle, the destination address may be a unicast address, an IP broadcast address, or a multicast group address. The set of security services offered by an SA depends on the security protocol selected, the SA mode, the endpoints of the SA, and on the election of optional services within the protocol. For example, one security protocol provides data origin  
25 authentication and connectionless integrity for IP datagrams.

The IP datagrams transmitted over an individual SA are afforded protection by exactly one security protocol. Sometimes a security policy may call for a combination of services for a particular traffic flow that is not achievable with a single SA. In such

From Original Application filed July 9, 2003

- security policy of the ordered list of Internet Protocol security policies. A particular security association entry based on the received packet is added to the ordered associative memory entries, the particular security association entry corresponding to the particular Internet Protocol security policy, and the particular security association entry being added
- 5 to the ordered associative memory entries prior to the particular associative memory entry location and after other security policy entries of the ordered list of Internet Protocol security policies located prior to the particular associative memory entry location.

4

From original Application filed July 9, 2003

don't care mask): (0x1, 0xe), (0x2, 0xd), (0x4, 0xb), and (0x8, 0x7). As shown, first, the TCAM entry d...d is checked to see if it matches a subset of the values covered by the range. If not, then the process is repeated with 0d...d and 1d...d. This happens recursively (using the stacks – not function recursion). Branches are trimmed when the entry being tested matches a disjoint set of values. Entries are saved when they match a subset of the values matched by the range. Entries that match overlapping sets are split and pushed onto the work stack.

FIG. 6A illustrates a process used in one embodiment for processing an inbound packet. Processing begins with process block 600, and proceeds to process block 602, wherein a packet is received. As determined in process block 604, if the packet is marked as conforming to IPsec, then in process block 606 the packet is processed, and processing is completed as indicated by process block 619. Otherwise, in process block 610, a lookup word is generated based on the received packet (e.g., with fields in accordance to those stored in the associative memory or other implementations of the data structure). In process block 612, a lookup operation is initiated and performed in the associative memory using the lookup word and a global mask register such that only SP entries are searched. The lookup result is received and a lookup operation based on the result is performed in the context memory in process block 614. Then, in process block 616, the packet is processed according to the action identified in the context memory. Processing is complete as indicated by process block 619.

FIG. 6B illustrates a process used in one embodiment for processing an outbound packet. Processing begins with process block 640, and proceeds to process block 642, wherein a packet is received. Next, in process block 644, a lookup word is generated based on the received packet. In process block 646, a lookup operation is initiated and performed in the associative memory using the lookup word and a global mask register such that both SP and SA entries are searched. The lookup result is received and a lookup operation based on the result is performed in the context memory in process block 648. As determined in process block 650, if the entry matched corresponds to an SA entry,